A High Performance Multi-standard Viterbi Decoder

Xuying Zhao, Huan Li
Institute of Automation, Chinese Academy of Sciences
University of Chinese Academy of Sciences
Beijing, China
{zhaoxuying2012 & lihuan2013}@ia.ac.cn

Xiaoqin Wang
Institute of Automation, Chinese Academy of Sciences
Beijing, China
xiaoqin.wang@ia.ac.cn

Abstract—**Convolutional encoder is widely applied in lots of wireless communication standards including 3G/4G mobile communications, DVB (Digital Video Broadcasting), IoT(Internet of Things) transmissions and so on. Therefore multi-standard Viterbi decoder design for the above receivers is a hot issue. In this paper, a reconfigurable high performance Viterbi decoder design is proposed for LTE, WiMAX, CDMA2000, GSM and TD-SCDMA. The proposed flexible architecture supports polynomial reconfiguration, constraint length of 5~9, and code rate of 1/2, 1/3, 1/4. Moreover, both tail-biting and zero trellis terminating modes are supported. To reduce decoding latency and complexity, it employs the forward traceback method and sliding window pipeline technology. Simulation results show that its maximum decoding throughput is 1.15Gbps under a clock frequency of 600MHz. Finally, its area is about 0.2mm^2, and the power consumption is about 46 mw, which is estimated with Synopsis DC and TSMC 28nm standard cell library.**

Keywords—viterbi decoder; multi-standard; high performance; forward traceback; sliding window

I. INTRODUCTION

With the popularity of portable data devices such as mobile phones, laptops and palm-sized computers, the data size for wireless communications shows an exponential growth. The future mobile communication systems (beyond 4G/5G) require for faster data transferring rate, better mobile traffic quality and more spectrum resource efficiency[1][2].

For wireless communication, there are problems such as attenuation, distortion, interference and noise in the transmission environment which leads to some errors in the received data[3]. Generally, such methods as CRC, automatic repeat request and convolutional encoding are used to detect and correct errors and of these methods the convolutional encoding using Viterbi algorithm has been widely used. Recently, the discussion on viterbi decoding applied in IoT (Internet of Things), MTC (Machine Type Communication) has also aroused widespread concern[4][5].

The viterbi algorithm based on maximum likelihood decoding was firstly proposed by Andrew J. Viterbi in 1967[6]. In the following years and even now, there are a lot of researches and designs for Viterbi decoder. However, quite a number of these studies aimed at a specific field[7,8,9,10,11,12,13]. And the computational performance of multi-standard supported Viterbi decoder can hardly satisfy the requirement of future mobile data traffic. For example, the data throughput of TI (Texas Instruments) 6670 VCP2 coprocessor is about 9.3Mbps under a clock frequency of 333MHz. As the computational performance is far from meeting the needs of mobile data transmission, there are four VCP2s in the SOC chip. In this paper, a high performance reconfigurable Viterbi decode for multi-standard receiver has been designed. The proposed Viterbi decoder has an architecture that supports the constraint length of 5~9, code rate of 1/2, 1/3 and 1/4. The generating polynomial can be configured flexibly and the tail-biting and zero trellis terminating modes are supported as well. It can be applied to wireless communication system (e.g. GSM, WiMAX, LTE) as well as other field such as UHF RFID (Ultra-High Frequency Radio Frequency Identification) receiver.

In the following, this paper is organized like these: Section Ⅱ explains the proposed algorithm and implementation scheme for reconfigurable Viterbi decoder. Synthesize results and performance evaluation are interpreted in the sectionⅢ and the final conclusion is given in section Ⅳ.

II. PROPOSED RECONFIGURABLE VITERBI DECODER

A. Viterbi decoding algorithm with forward traceback and sliding window methods

In this design, the forward traceback technique is adopted in the Viterbi decoder which is based on the unification property[14][15]. That is, if all survivor paths have been followed back *M* steps, they will converge to the same state, as is shown in Fig. 1. Typically, the traceback depth is set to 5~6 times the length of the constraint length[16].

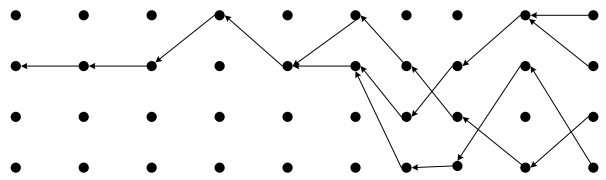

Fig. 1 unification property

Traditional register exchange method may cause amount of dynamic power consumption, so it should not be used in the situation of large constraint length; the decoding delay of backward traceback is linear related to the data length, so it is difficult to achieve high data throughput when the length is relatively large. With forward traceback technique, the data can be divided into several sliding

978-1-5090-3025-5/17/$31.00 ©2017 IEEE

windows and pipelining operation is carried out between the sliding windows. This kind of optimized algorithm gives attention to the “no delay” characteristic of the register exchange method and the “low power consumption” of the backward traceback method, which shows a great advantage in terms of implementation.

Forward traceback reconfigurable viterbi decoder function frame is shown in Fig. 2. LD and OUTPUT are data load/write back units; TBC is trellis branch compute unit; SPG is survival path generation unit and it includes a series of branch metric compute units and add-compare-select units; TB is trace back unit; CTR is the controller unit.

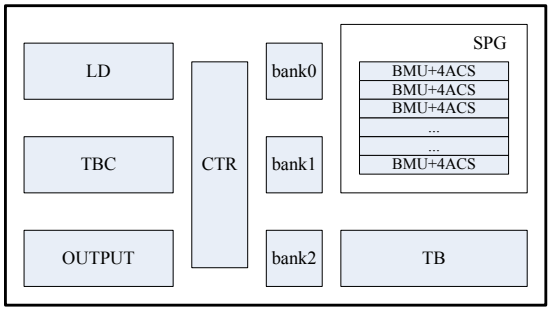

Fig. 2 reconfigurable viterbi decoder function frame

The proposed design carries out the pipeline mechanism with the unification property. As the maximum constraint length supported in this design is 9, the sliding window length is set to 64, reserving a certain redundancy to avoid performance degradation.

The principle of pipeline sequence design for Viterbi decoder is as follows: the backward traceback needs a forward recursive computation to gain initial trace back state of the previous sliding window, therefore, the SPG is made to one pipeline stage. Concrete pipeline sequence is shown in Fig. 3.

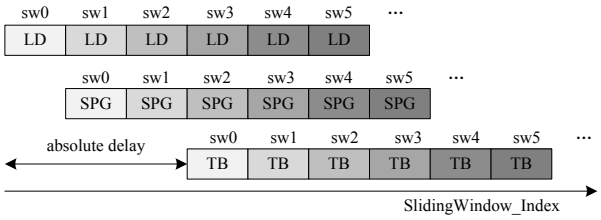

Fig. 3 Viterbi decoder pipeline sequence

From the above sequence graph, it can be seen that, after absolute delay of 3 sliding windows, the decoder begins to produce the result bits. Under the Radix-4 algorithm, 2 result bits are produced in each clock cycle. During the time interval [*kL*, *(k+1)L*], the target traceback state at time instant *kL* is obtained by the recursive computing of sliding window *k+1* . The target traceback state is the initial traceback state for TB operation.

TB operation is the traceback process. The state at time *t*, *t-2* is denoted respectively as s_t , s_{t-2} , and the survival bits at time interval [*t-2*,*t*] are denoted as d_t . The DC operation is defined in (1).

$$s_{t-2} = \{ s_t << 2, d_t \} \quad (1)$$

As the target start traceback state is obtained by forward recursive computing of the next sliding window, an extra sliding window is added at the end of the code block. The extra sliding window is obtained by repeating the original code block or using zero instead. The redundant computation of the added sliding window is hidden in the pipeline sequence, which will not bring extra time overhead. The simulation curve under Radix-4 algorithm is shown in Fig. 4.

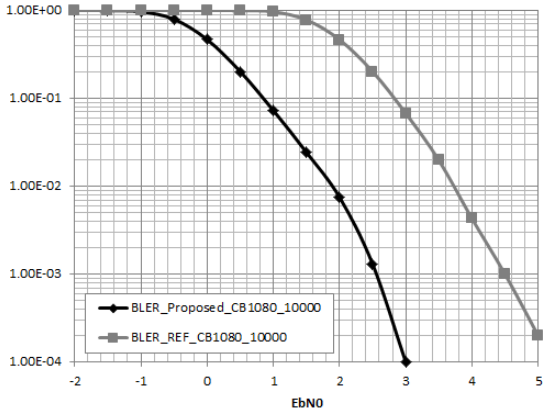

Fig. 4 BLER curve

In Fig. 4, the length of code block is 1080, and the number of code block is 10000. The channel model is AWGN, and the modulation scheme is QPSK. The BLER (Block Error Rate) performance gain of the proposed algorithm is about 1.8db better than the reference model which benefits from the added redundant sliding window.

B. Flexible Viterbi decoder architechture

The flexible architecture of Viterbi decoder is depicted in Fig. 5.

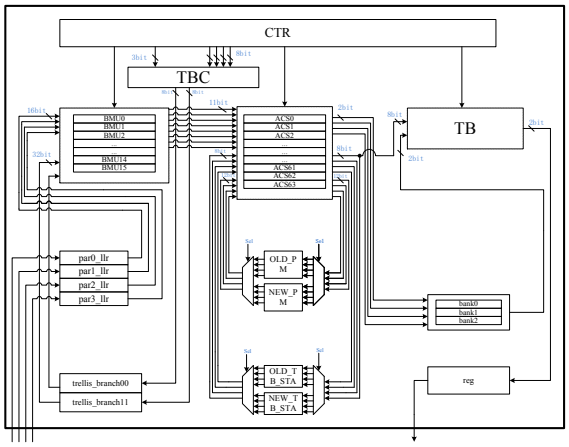

Fig. 5 the circuit architecture of Viterbi decoder

The difficulty in the implementation of Viterbi decoder lies in ACS unit. Four ACS units form a PE (Processing Element) unit. There are 64 ACS components in the proposed design. When the constraint length is 5, 6 and7, the number of states is 16, 32 and 64 respectively. All state

metric updating at a time can be completed in each clock cycle. When the constraint length is 8 and 9, the number of states is 128 and 256 respectively. The state metric updating at a time requires 2 cycles and 4 cycles respectively.

The design of PE is based on the state metric updating, which has the following regularity, as is shown in Fig. 6.

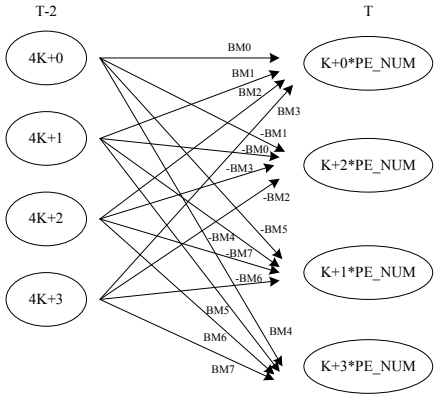

Fig. 6 processing unit regularity

In the Fig. 6, *K* is the PE index, *PE_NUM* is the PE number required by current configuration. Each state to be reached the next moment has 4 branches. Minimum or maximum state metric can be selected from the four branches, as a new state metric for the current state. This is the ACS process. The ACS unit structure under Radix-4 algorithm is shown in Fig. 7.

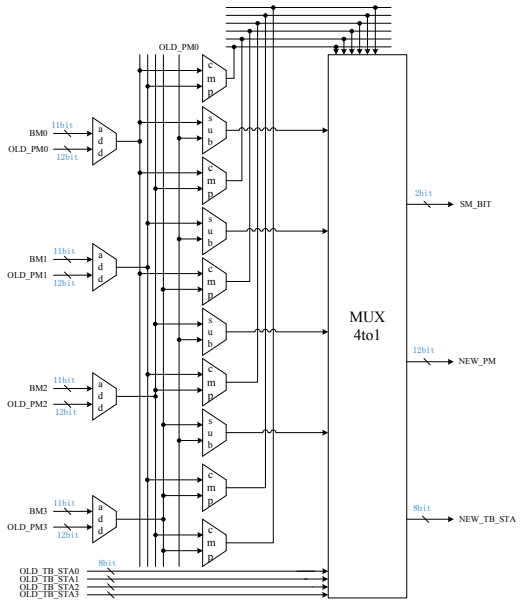

Fig. 7 ACS unit structure under Radix-4 algorithm

Considering area, the ACS unit structure under Radix-4 algorithm contains two more adders and two more comparators than Radix-2 algorithm. Considering time, one stage of comparison operation is added in Radix-4 algorithm. The logic cell of "choosing one from four" in this design contains six comparators, and the comparison result of four data is obtained in one clock cycle. At the same time the normalization operation joined which is hidden in the comparison operation. Normalization can prevent the data from overflow in the recursive computation and the critical path delay is reduced by the parallel computation.

The circuit structure of SPG process is shown in Fig. 8. The proposed design simplified the theoretically number of 1024 branch metrics to 8 by analyzing the internal rules between the state metrics and branch metrics. When the state metric is updated, it is required to record the branch which the current updated state is from. This branch contains two pieces of information: surviving bits and initial start state for traceback.

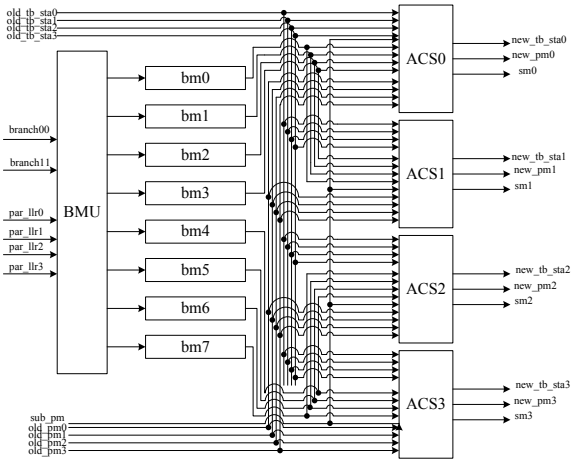

Fig. 8 the circuit structure of SPG

III. PERFORMANCE SIMULATION AND SYNTHESIS RESULTS

A. *Function simulation*

Function simulation and verification is accomplished by using the Synopsys VCS (Verilog Compile Simulator) and the UVM verification methodology. The simulation results are shown in Fig. 9.

B. *Synthesize*

This design adopted TSMC 28nm stand cell library and used Synopsys Design Compiler for synthesizing. The clock frequency is set to 600MHz and the operation voltage is 0.9v. Synthesize results are shown in table 1.

table 1 synthesize result

combinational cell count	107154	
sequential cell count	22010	
Design Area	211070um^2	
dynamic Power	44 mw	46mw
Leakage power	2mw	

C. *Throughput statistics*

Supposing the length of one code block is *K*, and the clock frequency is *f*. The throughput computation formula of Viterbi decoder under different constraint length is listed in the following table. The peak throughput of the decoder is given on the assumption that the code block length is 6144, and the clock frequency is 600MHz.

table 2 Throughput computation formula of Viterbi decoder under different constraint length

Constraint length	Throughput formula	Throughput（K=6144）
5	$\frac{2Kf}{224+K}$	1.15Gbps
6	$\frac{2Kf}{256+K}$	1.15Gbps
7	$\frac{2Kf}{320+K}$	1.14Gbps
8	$\frac{Kf}{288+K}$	573.13Mbps
9	$\frac{Kf}{544+2K}$	287.28Mbps

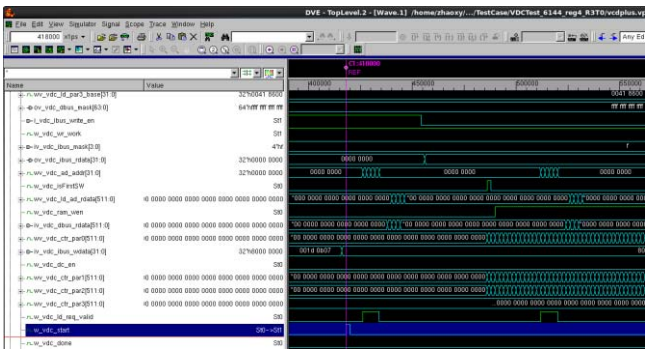

Fig. 9 VCS simulation

IV. CONCLUSION

This paper introduced a high performance multi-standard Viterbi decoder. The proposed flexible architecture supports polynomial reconfiguration, constraint length of 5~9, and code rate of 1/2, 1/3, 1/4. Both tail-biting and zero trellis terminating modes are supported. Moreover, forward traceback algorithm and sliding window pipeline technique are employed to improve the decoding throughput. It can be widely used in 2G/3G/4G wireless communications.

Potentially, this reconfigurable design can be applied to other wireless communication, such as DVB and IoT. Especially in some low power application scenarios, this Viterbi decoder can further reduce its power consumption by adjusting its clock frequency.

REFERENCES

[1] Khloud Mostafa, Aziza Hussein, Hassan Youness and Mohammed Moness, “High Performance Reconfigurable Viterbi Decoder Design for Multi-Standard Receiver,” 33rd national radio science conference, pp. 249 - 256, April 2016.

[2] N. Bhatt, M. Shah, and B. Asodariya, “FPGA Implementation of Power Efficient Low Latency Viterbi Decoder,” in International Journal of Engineering Research & Technology (IJERT), vol. 2, no. 5, May 2013.

[3] Mandwale, A. J., and A. O. Mulani. “Different Approaches For Implementation of Viterbi decoder on reconfigurable platform,” International Conference on Pervasive Computing IEEE, 2015.

[4] Tran T H, Kato H, Takamaeda-Yamazaki S, et al. “Performance evaluation of 802.11ah Viterbi decoder for IoT applications,” International Conference on Advanced Technologies for Communications (ATC), pp.320-325, 2015.

[5] Annamalai, Parthiban, J. Bapat, and D. Das. “Coverage Enhancement for MTC Devices using Reduced Search Viterbi Decoder Across RATs,” IEEE communication letters, vol. 20, no. 9, Sep. 2016.

[6] A. J. Viterbi, “Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm,” in IEEE Trans. Inform. Theory, vol. IT-13, pp. 260-269, Apr. 1967.

[7] Santhi M, Lakshminarayanan G, Sundaram R, et al. “Synchronous pipelined two-stage radix-4 200Mbps MB-OFDM UWB Viterbi decoder on FPGA,” Soc Design Conference. IEEE, 2009:468-471.

[8] Veshala, M., T. Padmaja, and K. Ghanta, “FPGA based design and implementation of modified Viterbi decoder for a Wi-Fi receiver,” Information & Communication Technologies, 2013:525-529.

[9] Liu X, Papaefthymiou M.C., “Design of a high-throughput low-power IS95 Viterbi decoder,” 2002:263-268.

[10] Wang H, Tan X, Chen F, et al. “A Viterbi decoder for UHF RFID digital baseband,” IEEE, International Conference on Asic, 2015.

[11] Gao B, Xiao Z, Chen Z, et al. “Multigigabit balanced add-select-register-compare viterbi decoders architecture in 60 GHz WPAN,” Communications. IEEE, 2013:531-535.

[12] Nargis J, Vaithiyanathan D, Seshasayanan R. “Design of high speed low power viterbi decoder for TCM system,” International Conference on Information Communication and Embedded Systems. 2013:185-190.

[13] Yoo W, Jung Y, Kim M Y, et al. “A pipelined 8-bit soft decision viterbi decoder for IEEE802.11ac WLAN systems,” IEEE Transactions on Consumer Electronics, 2012, 58(4):1162-1168.

[14] Gang Y, Erdogan A.T., Arslan T., “An Efficient Pre-Traceback Architecture for the Viterbi Decoder Targeting Wireless Communication Applications,” Circuits & Systems I Regular Papers IEEE Transactions on, 2006, 53(9):1918-1927.

[15] Cypher R, Shung C.B., “Generalized trace-back techniques for survivor memory management in the Viterbi algorithm,” Journal of VLSI Signal Processing Systems for Signal Image & Video Technology, 1993, 5(1):85-94.

[16] Manzoor R, Rafique A, Bajwa K.B., “VLSI Implementation of an Efficient Pre-Trace Back Approach for Viterbi Algorithm,” 2007:27-30.